Group Centric Networking: A New Approach for Wireless Multi-Hop Networking to Enable the Internet of Things

Greg Kuperman, Jun Sun, Bow-Nan Cheng, Patricia Deutsch, and Aradhana Narula-Tam
MIT Lincoln Laboratory
Lexington, MA, USA 02420
{gkuperman, jun.sun, bcheng, patricia.deutsch, arad}@ll.mit.edu

ABSTRACT

In this paper, we introduce a new networking architecture called Group Centric Networking (GCN) that is designed to support the large number of devices expected with the emergence of the Internet of Things. GCN is designed to enable these devices to operate collaboratively in a highly efficient and resilient fashion, while not sacrificing the users? ability to communicate with one another. We do a full protocol implementation of GCN in NS3, and verify GCN in emulation and on real hardware communicating over an RF channel. We show that GCN utilizes up to an order of magnitude fewer network resources than traditional wireless routing schemes, while providing superior connectivity and reliability.

1. INTRODUCTION

Despite decades of effort, multi-hop wireless networks have not succeeded in fulfilling their once-promised potential of providing ubiquitous connectivity with minimal fixed infrastructure. Today, almost all of our wireless devices are still tethered to wired infrastructure such as cell towers or 802.11 access points. But with the forecasted explosion in terms of users and data rates [1–3], having all devices directly connected to fixed infrastructure will no longer be tenable: wired access points will be overwhelmed and will quickly become bottlenecks in the network. If the concept of the Internet of Things [4, 5] is taken to its natural extent, then almost *everything* will be a “smart-object”, with all of these devices being wirelessly connected and exchanging data. Due to this impending surge of wireless devices, there has been a renewed focus on multi-hop networking to facilitate communications between these devices. The Internet Engineering Task Force (IETF) and preliminary 5G standards organizations have already begun putting forth ideas for designing future wireless systems, and multi-hop networking is a cornerstone for many of these next-generation architectures [6–9].

In this paper, we propose a new networking architecture called Group Centric Networking (GCN). The smart devices that will be deployed in these emerging networks will be resource limited and will be expected to operate in a lossy environment [10]. GCN is designed to enable these devices to operate collaboratively in a highly efficient and resilient fashion, while not sacrificing the users’ ability to communicate with one another. In particular, we design GCN to (1) efficiently handle the various types of traffic that future networks of smart-objects will carry, and to (2) take advantage of the wireless medium to resiliently connect the users of the network.

Although it is difficult to predict how future network traffic will behave, we can examine trends to make an informed guess. Most of today’s networks are *address*-centric, where one user typically acts as a client, and another as a server (e.g., your personal computer as the client, and a video-streaming service as the server). The server and client may potentially live anywhere in the network, and a routing protocol establishes an end-to-end route between the two.

In future wireless networks, the foundational concept of forming end-to-end routes between a client and server may no longer be appropriate. As others have [4, 7, 11] suggested, a potential future network might connect a large number of wireless smart-objects that are designed to work together in order to improve the quality of life for a human end-user, with these devices being located in some area local to that end-user. The users of this future network will have mostly localized traffic to communicate between one another, and any user can act as the source or sink of data in the network. We label any network with above characteristics as being *group*-centric, where the predominant traffic pattern is for data to be disseminated between a group of users in some local area that wish to operate collaboratively.

In particular, a group centric network has the following characteristics:

1. Devices will be grouped by an inherent set of “in-

This work is sponsored by Defense Advanced Research Projects Agency (DARPA) via Air Force contract #FA8721-05-C-0002. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

terests" that are dependent on the tasks they are performing, and these group members will wish to communicate reliably between one another. Devices are not limited to a single group, and can belong to multiple groups.

2. The majority of message exchanges will be within some local area, and long-distance traffic will only be a small fraction of overall communications.
3. Any device can be a source or a sink, and traffic patterns between them may be one-to-one, one-to-many, many-to-one, or many-to-many.
4. We hypothesize that a future wireless environment will have a mix of mobile and stationary devices, where mobility will be typically be limited to some local area.

An emerging example of a group centric network is a home or factory automation network, where various sensors, actuators, and controller systems work together to adjust to changing conditions in real-time [12, 13]. The devices in these networks will work together to ensure that environmental conditions are correct and that machinery is working properly to facilitate production. Another example of a group centric network that extends beyond groupings of low-power smart-devices is a military network, where both movement and communications are inherently localized. For movement, military operations are typically restricted to a certain geographic area, and for communications, a recent study shows that 95% of traffic in military networks travels at most three hops, with only 5% of traffic being long-range [14].

The design goal for Group Centric Networking (GCN) is to enable highly resilient and scalable communications between group members that are operating in a lossy, mobile environment. Current wireless networking schemes are ill-suited to meet these requirements. Today's approach for multi-hop wireless networking is to create end-to-end routes that are composed of a series of point-to-point links. These schemes are typically modifications of protocols that were designed for wired networks, and almost all wireless networking protocols (reactive, proactive, link state, distance vector, etc.) mimic this behavior to some degree. We believe that the characteristics of the wireless environment inherently make link-based routing unsuitable for wireless networking. The idea of a link is itself borrowed from wired networks: in a wireless network, there is no direct connection between two radios; transmissions are sent over-the-air and are typically overheard by multiple users. Any point-to-point wireless link is inherently unreliable due to interference, multi-path, and noise. The addition of mobility only exacerbates the situation. Since "links" in a wireless network are constantly fluctuating, constant route maintenance is required, which consumes significant network resources.

Group Centric Networking eschews links and routes in favor of a scheme designed specifically for the wireless medium. The key characteristics of the Group Centric Networking approach are:

- No link state or neighbor information is utilized or maintained, and minimal control information is exchanged.
- Data is efficiently disseminated only across the region where group members exist. To support this, we develop a novel Group Discovery algorithm that dynamically discovers the region of interest and efficiently selects the minimal amount of relay nodes required to "cover" this region.
- Reliable communications is achieved in an error-prone and mobile environment by using "tunable resiliency", where the number of redundant data relays is configurable and is able to self-adjust in response to real-time channel conditions.
- Devices communicate in a many-to-many traffic pattern. Efficient one-to-one, one-to-many, and many-to-one are subsets.

As we will demonstrate, GCN utilizes up to an order of magnitude fewer network resources than traditional wireless routing schemes, while providing superior connectivity and reliability.

We verify our approach by implementing the full set of protocols for Group Centric Networking in NS3 Direct Code Execution (DCE) [15, 16], which allows for a high-fidelity comparison against other wireless networking protocols, and enables an easier transition of GCN protocols to other researchers in the community. The results were verified in the real-time emulation environment EMANE/CORE [17, 18], and basic tests were performed on actual hardware devices communicating over an RF channel.

The outline of this paper is as follows. To further motivate the need for a new approach for wireless multi-hop networking, in Section 2 we provide results and analysis on the limitations of current wireless networking schemes. In Section 3, we present Group Centric Networking and its major mechanisms. In Section 4, we discuss our implementation of GCN in NS3 and EMANE, and present simulation results demonstrating the performance of GCN. In Section 5, we conclude and discuss ongoing work and future directions for GCN.

2. LIMITATIONS OF CURRENT APPROACHES FOR WIRELESS NETWORKING

In this section, we further motivate the need for a new approach to wireless multi-hop networking by demonstrating the limitations of current wireless networking schemes. Current wireless networking schemes can be grouped into two general categories: link-based, where a

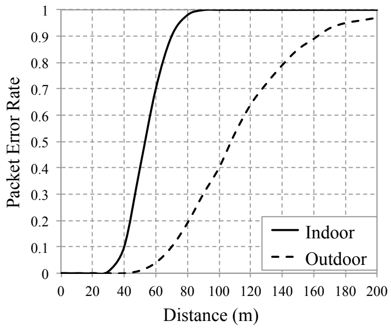


Figure 1: Packet error rate curves for IEEE 802.15.4 devices [20]

route is formed by series of point-to-point links, or flood-based, where a packet is continuously retransmitted by almost every user in the network until the intended destination is reached. Since the bulk of current and proposed wireless networking protocols are still link-based, we focus our attention on link-based schemes. In this section we present simulation results for link-based routing that demonstrate the following: (1) in a wireless environment with nearly ideal conditions (zero errors for short range transmissions and no mobility), link-based routing has high loss due to the inherently unreliable nature of link-state control information; (2) in a lossy environment, link-based routing is incapable of reliably delivering data messages; and (3) the route repair and maintenance process that link-based routing schemes employ to overcome lossy links is both costly and unable to ensure reliable data message delivery.

The basic mechanism of link-based routing schemes is the broadcasting of a control message (called a “hello”) to all of its neighbors. If a response is received, a link exists between the user and neighbor, and routes are then formed on top of these links. When the link is reliable and doesn’t change frequently, this scheme works well. However, with dynamic and variable links, current link-state information may be stale or indicate the presence of a link that has a very high error rate. To help mitigate the potential issues arising from links not being reliable, a number of approaches have been proposed, with ETX [19] being perhaps the most well-known of these schemes. ETX tries to assign a cost to a link that is proportional to that link’s reliability.

When modeling the wireless channel, researchers often use a strict cut-off for transmission distance: any device within a certain distance of the transmitter will receive the message, and any device beyond that distance will not. Even if we assume there are no other active transmitters that can be potentially interfering, this is not a realistic model for a wireless channel: there is no strict cut-off. The effects of multi-path, thermal fluctuations, and other random variations of the environment will induce a “transition region” in which the probability of packet error increases from 0 to 1.

For our wireless channel model, we use packet error rate (PER) curves for IEEE 802.15.4 devices from [20] (reproduced in Figure 1). These curves were determined through both simulations and hardware measurements. The authors of [20] determined two curves: one for indoor transmissions and one for outdoor. For our simulations, we use the indoor curve which has a sharper transition region and more closely approximates the strict cut-off range that is typically used.

This curve assumes no loss for short range transmissions. However, in the presence of interfering wireless devices, one would not expect 0% packet loss for transmissions at close range. Various papers have tried to quantify the effects of interference on packet reception rates for devices operating in the 2.4 GHz ISM band (where 802.15.4 operates) [20–22]. These studies find that loss can be on the order of 25%, if not greater. We define a new curve for a higher loss environment where the minimum PER is 25% for short range transmissions. The PER curve to model interference that causes 25% packet loss is constructed by multiplying the packet success rate at any given distance d (i.e., $(1 - \mathrm{PER}(d))$) from the indoor curve of Figure 1 by $(1 - 0.25)$. We label the two PER curves as `Curve 0%` and `Curve 25%`.

Additionally, we compare the effect of the transition region with the more traditional transmission model often used in literature that assumes a user has a fixed transmission range, where within that range all transmissions are successful. This fixed-distance error curve has the following parameters: a transmission under 40 meters has 0% PER (100% reception), and a transmission over 40 meters has 100% PER. We label the curve with a fixed transmission distance as `Fixed`.

We believe that link-based routing protocols are particularly vulnerable to the effects of the transition region. Control packets will occasionally be successfully exchanged by users that are a far distance apart, which will lead to poor quality links being selected for routes. These long-distance links will typically be preferred over shorter, more reliable links in a shortest path routing protocol. Routes continue to use the long-distance link until the link timeout period expires.

For our simulations, we evaluate the Ad Hoc On-Demand Distance Vector (AODV) routing protocol [23]. AODV is used in the ZigBee multi-hop networking standard [24], and is the basis for new proposals to connect networks of smart-objects [25, 26]. We operate AODV in “standard” mode, which is the routing protocol with its default parameters, and in “ETX” mode, which has ETX metrics enabled. To the best of our knowledge, there has been no extensive characterization of wireless routing protocols operating in the presence of a transition region with respect to packet error rates.

We consider the following simulation scenario: 25 nodes are randomly distributed in a circular area with

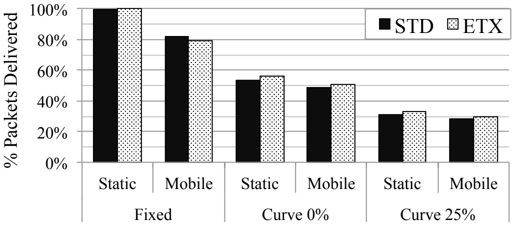


Figure 2: AODV: Packet delivery success rate

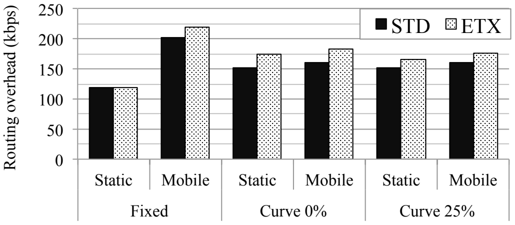


Figure 3: AODV: Routing overhead

a radius of 100 meters. Starting 30 seconds into the simulation, each node sends one 64 byte packet per second to every other node. The test is run for a total of 30 minutes (1800 seconds) of simulation time. There is no mobility in the network. Simulations are run using the OPNET network modeler [27], which has AODV implemented according to IETF standards.

We first examine the transition region's affect on packet delivery rates. In Figure 2, we plot the percentage of packets that were successfully received at the destination. For AODV, we see that in the supposedly ideal environment of no mobility and no interference (`Curve 0%`), packet delivery rates reach only a maximum of 56% for ETX, and only 53% for standard. This is in contrast to the fixed transmission model that has 100% packet reception. When the wireless channel includes interference and becomes lossy at short ranges, we see that packet delivery rates plummet. With `Curve 25%`, AODV is only able to deliver only about 32% of packets for either variant.

Next, we examine the amount of network resources used to find and maintain routes between users. The bandwidth occupied by AODV routing control messaging is plotted in Figure 3. As a baseline, we consider the overhead generated when the `Fixed` error curve is used. AODV baseline using `Fixed` produces 119 kbps of control traffic for both standard and ETX. AODV has a route repair procedure, where upon detection of a change for a route, new control messaging is exchanged to find an alternative path. If links are changing frequently, then more control traffic will be generated than if links were rarely changing. In the presence of a transition region, AODV with ETX believes routes are changing frequently, and routing overhead goes to 150 kbps for standard AODV and 175 kbps when ETX is enabled. In bandwidth limited environments, this level of background traffic may congest the network and not allow data messages to be transmitted. Additionally, high levels of control traffic causes a device to transmit more frequently, which increases interference and can consume significant battery power.

Ultimately, one needs to examine how much is gained by the high level of control messaging that is used to find and maintain link-based routes. Using periodic control messages to determine which links are valid in a wireless network leads to many poor quality links being active for long periods of time. This in turn causes poor quality routes to be formed between users, which leads to high packet loss. Schemes meant to improve the quality of links chosen for routes, such as ETX, do not appear to substantially improve performance.

3. GROUP CENTRIC NETWORKING

In this section, we present the core mechanisms that form Group Centric Networking (GCN). GCN is designed to efficiently and robustly support groups of users desiring to communicate with one another in a localized region. Many new emerging wireless devices will be resource limited and will be expected to operate in a lossy environment. Communication protocols must be (1) resilient against packet errors due to interference and mobility, (2) bandwidth and power efficient.

We define a group to be a collection of users that regularly communicate with one another. GCN enables a set of users to efficiently and resiliently communicate with any other set of users in a group via a many-to-many traffic pattern. One user may wish to disseminate information to the entire group or to only some members of that group. Alternatively, some set of users may wish to collect information from another set of group nodes. The "many" of the many-to-many traffic pattern can either be some or all of the group users. One-to-one, one-to-many, and many-to-one are all considered subsets of the many-to-many traffic pattern.

An example layout for a group centric network is shown in Figure 4. A set of relay nodes has been activated such that all group members are connected to one another. There are multiple opportunities to overhear a message in case of packet loss, and the failure of any individual relay or link will not prevent messages from being received by other users. User a may wish to communicate to the entire group, or just to users b and c. Both of these communication types are efficiently enabled by the one-to-many traffic pattern. Alternatively, a may wish to receive data from users b and c via a many-to-one message. While only one group is shown, users can belong to any number of groups.

To achieve our design goals of scalable, efficient, and resilient group communications, the major mechanisms of Group Centric Networking are as follows:

1. *Group discovery*: Efficient discovery of the local region where group members reside via a a group

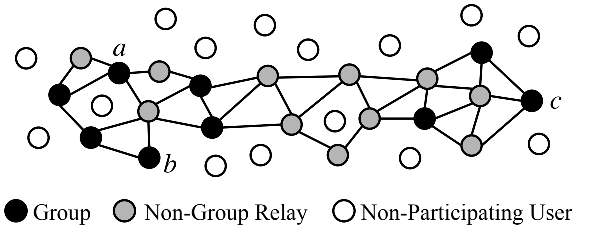


Figure 4: An example of a group centric network

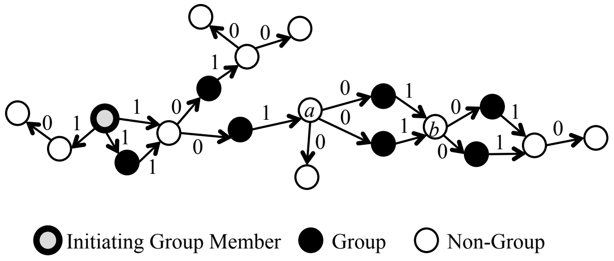


Figure 5: Discovering the local region using discovery regeneration. Each arrow shows the time-to-live (TTL) of the outgoing discovery message.

discovery algorithm that is able to connect group members without the use of global control information. Group Discovery is discussed in Section 3.1.

2. *Tunable resiliency*: Relay nodes are activated such that the local region is sufficiently "covered" in data by having a tunable number of redundant data relays. This allows for resiliency towards both packet loss and mobility without the need for the constant exchange of control information. The number of activated relay nodes self-adjusts in response to real-time channel conditions. Tunable resiliency is described in Section 3.2.

3. *Targeted flooding*: Data can be efficiently and resiliently sent between sets of group members through an approach we call "targeted flooding". The mechanism for targeted flooding is detailed in Section 3.3.

3.1 Group Discovery

The purpose of group discovery is to find and connect group members in a local region without prior knowledge of where those group members reside, and to do so efficiently without globally flooding control messages. A naive approach would be to use a control message for discovery that has some time-to-live[1] (TTL) set to the maximum number of hops the group is expected to extend from end-to-end. The discovery message is transmitted across the network, with the TTL being decremented at each next user. While the message would reach the entire group, it would also travel into areas where group users do not exist. In a large network with limited bandwidth, this can be a significant waste of network resources.

For group discovery, we introduce a novel approach that we label *discovery regeneration*, where a group discovery message is regenerated with some small "source" TTL by each group member. By doing so, the reach of the discovery message is limited to some fixed distance around the local region where group members live. The basic mechanism for Group Discovery using discovery regeneration is as follows.

- A group member initiates group discovery by sending out a discovery message. The TTL for the discovery message will be set according to how far around any particular group member the discovery region is to extend. We refer to the TTL that the initiating user sets as the source TTL.

- If a group member hears a discovery message, it will regenerate the message with the source TTL.

- If a non-group member hears a discovery message with a TTL greater than zero, it will decrement the TTL, and rebroadcast the message. If a non-group member hears a discovery message with a TTL equal to zero, it does nothing.

When group members receive a discovery message, it sends back an acknowledgment (ACK) to the previous group node that relayed the advertisement. All nodes in between the group nodes that receive the ACK are elected as data relays. If multiple discovery messages are heard, the ACK is sent only to the neighbor that sent the first one. Duplicate detection is used by all users in the network to ensure discovery messages are broadcast only once. Note that ACKs are only sent to the group node that regenerated the source TTL, and not to the source of the initial discovery message. This approach is different from traditional multicast whereby *join* messages are sent to the root of the tree. In GCN, when a user becomes activated as a relay, it is now a relay for the entire group, and not for any particular group node. Relays do not need to maintain any information on who sent it an ACK. Similarly, a group user is listening for data from any relay, and the group user does not need to remember which relay nodes it activated. After the group discovery process is complete, no link-state or neighbor information is maintained by any user in GCN.

[1] Time-to-live (TTL) is a field used in data packets to limit the distance a packet travels. Each time a packet is retransmitted, the TTL gets decremented by one, and once the TTL reaches zero, the packet is dropped.

An example of group discovery with regeneration is presented in Figure 5. We first discuss the dissemination of the discovery messages. Each arrow shows the time-to-live (TTL) of the outgoing discovery message. Group members regenerate the TTL at the source value of 2 (which is decremented to 1 at transmission). Non-group members do not regenerate the TTL, which limits the reach of the discovery message to the local region where group members live. All group members are discovered without the need to have control information extending beyond the local area. Next, we discuss how users are selected as relays using acknowledgment messages. The non-group user a has two group members that hear its discovery message. Each of those group members send back an ACK to a, but once a receives one ACK, it becomes activated as a group relay node and can ignore any additional ACKs that it hears. Non-group user b hears a discovery message from two group members. User b will choose one of the two to send an ACK; by default it chooses the first user it hears a discovery message from. Hence, only one of the two group members that b heard a discovery message from will be activated as a relay.

3.1.1 Effect of Regenerated TTL on Group Reach

In order to discover all group members, the regenerated TTL value needs to be sufficiently high such that all users are within the discovery region. Setting the TTL too high, however, will result in wasted transmissions. To quantify a recommended TTL value, we perform analysis and simulation to show that low values of TTL are sufficient to discover all group members, even if the group is sparsely populated. For the analytic model, we develop a first order approximation that predicts the number of group users that will be discovered as a function of the source TTL. Due to space constraints, we only the present the result of our analytic model. We consider N users that are uniformly distributed across a two-dimensional region with an area of A; the density of users is given by $\lambda = \frac{N}{A}$. A user in this region is a group member with probability P_g, and each user has a transmit distance of X. Given this set of assumptions, our approximation for the the expected percentage of group members that are discovered with a source TTL of T is $1 - e^{P_g \lambda \pi ((X - \frac{1}{2\lambda}) \cdot T)^2}$.

In addition to the analytic model, we performed a simulation using our implementation of GCN in NS3 and compare the results to what is predicted by the analytic model. For the simulation, we consider 100 users uniformly distributed in a circular region with a radius of 100 meters, and a transmit distance of 40 meters per user. We test three different group densities, where a user is a group member with a probability P_g of either 5%, 10%, or 25%. All users are stationary for the duration of the test, and a group member is selected at random to initiate the group discovery process. The source TTL is varied between 1 and 4.

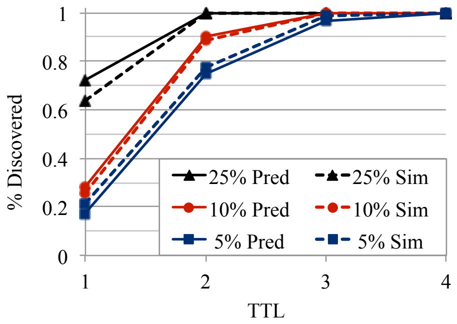


Figure 6: Predicted and actual percentage of group members found during the discovery process as a function of source TTL.

Figure 6 shows the average simulation and predicted results over 50 random seed runs. First, we observe that for low values of group membership probability, which leads to group members being far apart, low values of source TTL are sufficient to find all users. For a group probability of 5% (where we expect only five group members on average), a source TTL of 3 allows 98.6% of the group to be found on average. For a group probability of 25%, a source TTL of 2 finds 99.8% of group members. Next, we observe that the analytic model is a close fit for the results from the NS3 simulation. Thus, a user can use the analytic model to select an appropriate source TTL for efficient discovery.

Once the group discovery process is complete, an efficient set of relays has been activated across the local region to connect all of the group members together. Messages can now be sent from any user to the entire group via a one-to-all traffic pattern, which will form the backbone of the many-to-many traffic pattern (presented in Section 3.3). The frequency of group discovery messages and the number of relays needed to provide resilient coverage are functions of network dynamism (i.e., due to mobility, loss, joins/leaves, etc.).

3.1.2 Total Transmission Comparison

To understand the savings of transmissions sent over-the-air with GCN (for both data and overhead), we compare it to two different methods for dissemination of data in wireless networks. GCN is designed for messaging in a local region; hence, we compare against Simplified Multicast Forwarding (SMF) [28], which floods a local region with data while employing duplicate packet detection to limit retransmissions. In SMF, a message is transmitted with some TTL, and that message is then continually rebroadcast by each subsequent user until the TTL expires. No control messaging is required in SMF, and there is no mechanism to dynamically set the TTL.

To offer a fair comparison against GCN, when we op-

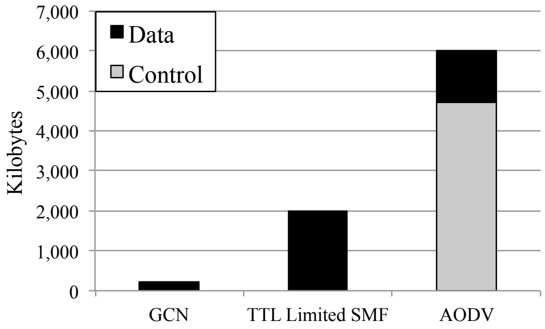


Figure 7: Total bytes sent over-the-air to transmit all data packets from a source to the entire group.

erate SMF, we assign a message the minimum TTL for it to reach all of the group members. We also compare GCN against the Ad-Hoc On-Demand Distance Vector (AODV) routing protocol [23], which finds a route from a source to a destination at the time a message is to be sent. We note that there exists a multicast version of AODV, called Multicast AODV (MAODV) [?], but there is no implementation available to compare against. In MAODV, each multicast group member requires a unicast route back to the source, and to find these unicast routes, MAODV uses AODV control messaging. Hence, MAODV should utilize the same amount of control information as AODV.

In our simulation, group nodes are operating in a local region as consistent with a personal area network. We consider two concentric circles, one with a radius of 100 m, and the other with a radius of 200 m. Group members reside within the smaller circle, which we denote as the local region. Users are uniformly distributed across the entire network, with 400 users total and 100 users in the local region. Each user has a transmit distance of 40 m. Of the users in the local region, 10% are group members (on average, we expect 10 group members). All users are stationary for the duration of the test. One group member is randomly selected as the source. This source will initiate group discovery, and then send a data packet destined to all members of the group at the rate of 1 packet per second for 10 seconds. A data packet is 1400 bytes. In GCN, a group discovery message is 14 bytes and an acknowledgment packet is 20 bytes. AODV is run with its default parameters.

Figure 7 illustrates the average of 50 random seed runs totaling the traffic sent over-the-air from a single source to the entire group. As can be seen, GCN is able to efficiently discover the local region, and find the minimum number of relays necessary to disseminate the set of packets to the entire group. GCN transmits a total of 220 KB over-the-air, with only 6.5 KB of that being control information. TTL limited SMF requires 2,001 KB to disseminate these same 10 packets. SMF does not discover the local region, and hence floods data into areas where group members do not exist. This is particularly problematic when the source node is located at the edge of the group. AODV transmits a total of 6,000 KB to disseminate the data to the entire group, with 4,700 KB of that total being control traffic. As mentioned earlier, AODV is not a multicast protocol, hence it sends a separate copy of the same packet to each user. But even if we reduce the data portion by a factor of 10, the control information sent over-the-air is still significantly greater than what either GCN or SMF use in total.

3.2 Tunable Resiliency

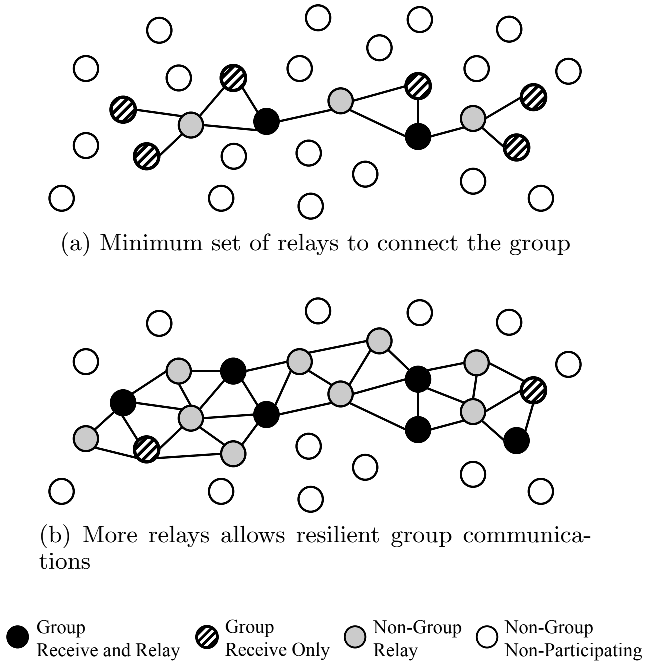
As noted above, group discovery activates an efficient set of relays (i.e., greedily selecting the minimal set) such that all group members are connected. This immediately enables the one-to-many traffic pattern: when a user transmits a packet to the group, all group members will now be able to receive it. But, this minimal set of relays is not particularly robust for group-wide dissemination as a single packet failure can cause all downstream group members to not receive the data. Also, if a set of users move such that they are no longer in range of a relay, or if a set of relays move out of range of one another, then the group can become disconnected.

To make GCN more robust, we extend group discovery by adding a mechanism we call *tunable resiliency*. Tunable resiliency allows for the targeted activation of additional relays to provide sufficient coverage in order to protect against packet loss and mobility, and to have those relay nodes be able to self-adjust and adapt to the current network conditions. The number of activated relays is able to self-adjust to respond to real-time channel conditions. To enable tunable resiliency, the group discovery process is extended as follows:

1. A short delay is added to the discovery acknowledgment (ACK) messages.
2. Each user keeps a count of how many neighbors it sees in order to determine the number of possible data relays within that user's neighborhood.
3. A set of users will self-select as data relays in a probabilistic fashion to achieve the desired density of relays to enable robust data coverage. This is in addition to the set of users that are selected as data relays through the group discovery process.

The purpose of the short delay before transmitting an ACK is to allow discovery messages to propagate through the immediate vicinity of a particular user. Discovery messages are retransmitted as soon as they are received.

For the neighbor counting process, a user will count the number of discovery messages that it hears from other users in its immediate vicinity. When a user receives a discovery message, it then immediately retransmits that message (unless that user is a non-group


(a) Minimum set of relays to connect the group

(b) More relays allows resilient group communications

Figure 8: Change in coverage using tunable resiliency.

member and the discovery message has a TTL of zero). The neighboring users will receive the discovery message, and immediately rebroadcast it themselves. Since the discovery messages are transmitted immediately, after a short amount of time, a user should be able to count the number of discovery messages it has heard. This allows the user to get an estimate for the number of users that are within its neighborhood.

By having an estimate on the number of users in a neighborhood, nodes can now self-select as data relays to achieve the desired density for data coverage. Assume that we wish to have R data relays within range of any interested user. The value R specifies the density of data relays for the group, and higher values of R provide additional resilience against packet loss and mobility. Recall that in the group discovery process, an ACK is addressed to a particular user to activate it as a relay. We call this user the *obligate* relay. If a user is specified as an obligate in an ACK message, it will always become a relay. To allow users to self-select as relays, a field is added to the ACK that specifies a probability of accept (ACP). If a user receives an ACK and it is not the obligate, it then becomes a relay with probability ACP. Once a user becomes a relay (either by being the obligate or by self-selecting), it then continues the discovery process by sending a new ACK that follows the same steps as above.

The ACP value is set as follows. Assume that a user has counted N neighbors and desires to have a total of R data relays within its range. The first node that user heard a discovery message from will be selected as the obligate relay, and the ACP value will be set to $\frac{R-1}{N-1}$.

This approach for probabilistically selecting data relays allows the network to self-adjust to real-time error conditions. The number of discovery messages heard by each user reflects the current error rate being experienced in the network. For example, assume there is a 50% packet error rate due to interference or some other loss; if ten neighbors of user U transmit a discovery message, then on average five of those messages should be expected to be heard by U. If we assume a wireless channel has a similar error rate in both directions, then an ACK sent by U should reach a similar number of neighbors that U initially heard a discovery message from (i.e., about five of the ten neighbors should hear an ACK). If U desired to have three data relays in its vicinity, it will send an ACK with one obligate and an ACP set to $\frac{2}{4}$. On average, this will activate close to the three desired relays.

We add an additional requirement to have tunable resiliency function as desired. To maintain a uniform distribution of relay nodes across the group region, a user will only attempt to self-select once; i.e., the user will not attempt to self-select a relay with each subsequent ACK it receives. If that user is specified as an obligate in any ACK, it *will* become a relay.

After an iteration of group discovery with tunable resiliency has been performed, the area where the group resides will have a sufficient density of relay nodes to increase data coverage and become resilient to loss and mobility. In Figure 8, the change in coverage is shown by use of tunable resiliency. In Fig. 8a, the minimum set of relays is activated and the entire group is connected. If any group member sends a one-to-many transmission, all users will receive the message. But, if any packet is lost, or any relay moves out of range, then the group will become disconnected. In Fig. 8b, additional relays have been activated by setting the ACP in the ACK message to activate the desired number of relays. This allows data to cover more of the group area, which increases the resiliency of the group against packet loss and mobility.

To evaluate the effect that tunable resiliency has on Group Centric Networking, we look at two criteria: (1) the connectivity of a group when users are mobile, and (2) how reliably and efficiently messages can be delivered in the presence of packet loss and mobility. The desired number of data relays that any user wants to activate is denoted by R.

3.2.1 Effect of Mobility on Group Connectivity

For the first test, we consider 100 users uniformly distributed in a circular region with a radius of 100 meters. Any user in this region can be a group member with a probability of 25%. The number of desired data relays R is set to 2. Users move according the the random waypoint model, with a speed of 0 to 5 m/s and a pause time of 0 to 2 seconds. The test is run for 1000

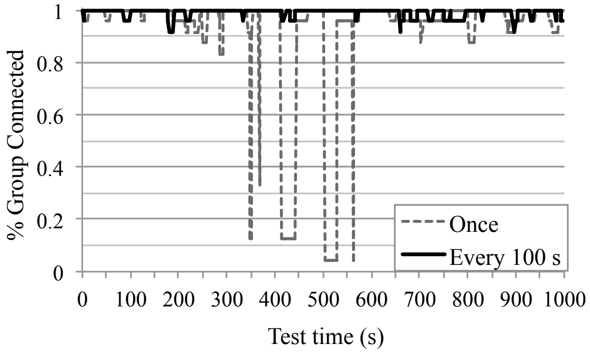


Figure 9: Group connectivity with mobile users.

seconds. One group member is randomly picked as the source, and it will initiate group discovery. The source then either (1) never initiates another group discovery for the remainder of the test, or (2) it initiates one every 100 seconds. To measure group connectivity, we sample the network every second and determine if there exists a path from the source to each group member.

In Figure 9, we plot the percentage of group members that are connected to the source as a function of time. When a group discovery message is sent every 100 seconds, the group has very high connectivity. Overall, the the source has a direct connection 99% of the time to any other group member. Next, we see that even when only a single group discovery is performed, the group remains highly connected for long periods of time. With only a single group discovery, the source has a connection to any other group member for 92% of the 1000 second test.

3.2.2 *Effect of Resilience Factor in Lossy Environments*

Next, we examine the effect of tunable resiliency on the reliability of message delivery in the presence of packet loss and mobility. We use the packet error rate (PER) curves introduced in Section 2: `Curve 0%` and `Curve 25%`. Recall that these curves have a transition region where the PER goes from some base level to 1. `Curve 0%` has a base level PER of 0% (no loss at shorter distances) and `Curve 25%` has a base level PER of 25%. We introduce another PER curve for a highly lossy environment that has a base level PER of 50%. We label this `Curve 50%`, and it is constructed in the same fashion as `Curve 25%`.

The following simulation is performed. 100 users are uniformly distributed in a circular region with a radius of 100 meters. Any user in this region is a group member with probability of 25%. Transmission distances and error rates are given by the PER curve that is being used for that particular test. Users are either stationary or move according the the random waypoint model with a constant speed of 2 m/s. One group member is selected at random to initiate group discovery. The number of desired data relays R is set to either 1, 3, or 5. When $R = 1$, only an obligate relay will be selected, and tunable resiliency is effectively turned off. Each group member sends a packet to the group once every second for 100 seconds. The number of packets successfully received at each group member is recorded. Additionally, the total number of bytes sent over-the-air by all users is recorded. Similar to the test performed in Section 3.1, we compare against SMF, which floods the data across the entire region. Fifty random tests are run, with the results averaged.

The packet delivery success rate is plotted in Figure 10, and the total traffic sent over-the-air is plotted in Figure 11. For $R = 1$, which does not use tunable resiliency, packet delivery rates are low, but not as low as one might expect. This is because different group members that are close together will not all select the same user to be their relay, and multiple data relays will become activated within a neighborhood. This allows for a small level of additional resiliency against loss. As the target number of data relays per user R increases, so does that packet delivery rate. Without mobility, setting R to 3 allows packet completion rates above 94% for both `Curve 0%` and `Curve 25%`. With $R = 5$, all tests under all three curves have packet completion rates exceeding 96%. This matches the same resiliency that is provided by flooding the data across the entire region as is done by SMF.

Next, we look at the total bytes that were sent over-the-air to deliver these packets (Figure 11). By selectively activating additional relays throughout the group area, tunable resiliency allows for higher resiliency without consuming significant network resources. Flooding the data across the region consumes significant resources, and does not provide substantial benefit in terms of delivery rates over GCN with tunable resiliency. We note that the number of bytes transmitted by SMF decreases as packet error rate goes up. The reason for this is that when there are many packets in error, there are fewer packets to retransmit. We observe that, for a given value of R, GCN maintains a fairly constant level of bytes transmitted over-the-air for the different error curves. This is because the selection of data relays is able to self-adjust to respond to the channel conditions, and activate a constant number of relays regardless of the error rate being experienced.

3.3 Targeted Flooding

In the previous sections, we described how Group Centric Networking discovers group members and forms a resilient one-to-all communication pattern between all of them by use of tunable resiliency. But sending all messages to the entire group is not always efficient. For example, a group of sensors may want to send data to a single data collector via a many-to-one traffic pattern. Alternatively, some group member may want to

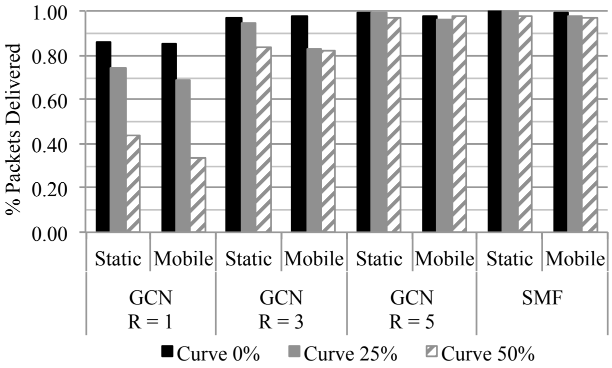


Figure 10: Packet delivery rates of GCN and SMF

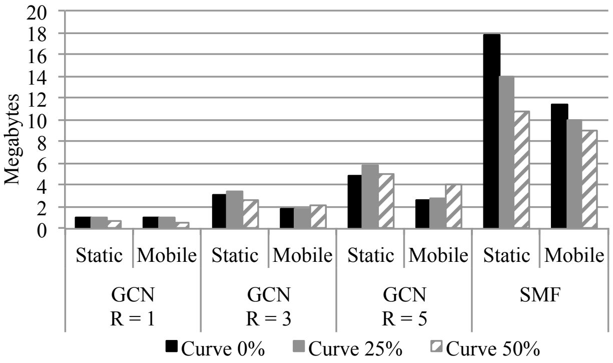


Figure 11: Number of bytes sent over-the-air with of GCN and SMF

query a subset of users, or have a one-to-one communication with one particular group member. We wish to enable these additional traffic patterns in a resilient manner without requiring additional control data to be sent throughout the network. In this section, we present a mechanism for robustly sending data to a specific set of group members through a process we call *targeted flooding*.

To be able to target transmissions towards specific users, targeted flooding uses distance information gathered from overheard packets to create a distributed gradient field towards each of the group members. Each transmitted packet (data or control) will be tagged with the originating user's ID, and a counter will be attached to that ID that indicates how many hops this particular packet has traversed. Each time a packet is retransmitted, the counter is incremented. When a user hears a packet transmission, it records its distance to the source user. This adds a minimal amount of overhead to each packet. Using the distance information collected, a user can transmit a packet destined to another group member without knowing anything about available links, or even who its own neighbors are. To help facilitate a more robust gradient towards any user, intermediate relays can add their ID to each retransmitted packet, allowing users to more quickly learn how far they are from other users, not just the initial source of packet.

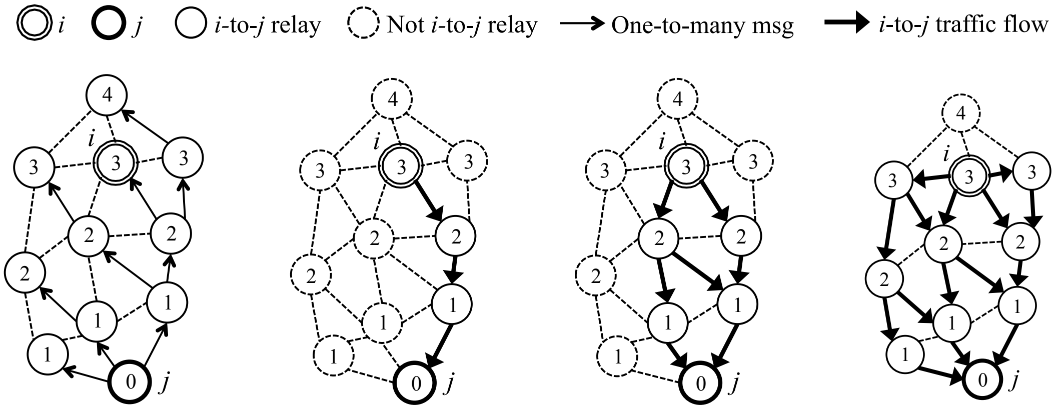
In the following subsections, we describe how one-to-one, one-to-many, and many-to-many traffic patterns can be supported with GCN.

3.3.1 One-to-One Traffic Pattern

For some user i, we label its recorded distance to user j as Δ^i_j (i.e., if i believes it is four hops from j, $\Delta^i_j = 4$). A packet destined for a specific user will have two fields in its header: a destination, and a maximum retransmit distance (MRD). When a relay node hears a packet with a particular destination, it looks at that packet's MRD value, and if that value is greater than or equal to its own distance from the destination, it will rebroadcast the packet with the MRD field decremented by one. In other words, if user i receives a packet destined for j with MRD $\leq \Delta^i_j$, i will retransmit the message with MRD $= \Delta^i_j - 1$.

This approach will allow a packet to flood a narrow corridor towards some particular destination. The width of the corridor that a packet travels can be modified by changing the MRD value at the originating user. A higher value for MRD will cause a packet to spread farther around the source, which causes a wider set of paths to be traversed as it funnels towards the destination. By changing the value of MRD at the source user, we can tune the number of relays that are used to relay data for a one-to-one flow. Hence, the resiliency for one-to-one traffic is tunable to allow for additional robustness. Because the message is flooded through a corridor towards the destination, the distance information needs to only be "good enough". Each time a new message is overheard by some user, the local distance information will be updated. This process allows for a constant refresh of distance information without the need for dedicated control information.

We demonstrate the one-to-one flow via targeted flooding through the example in Figure 12. In Figure 12a, user j sends a message via one-to-all to the entire group (this could be a group discovery initiated by j); all users learn their distance from j. In Figure 12b, we show the naive approach of only using the reverse path that the message traveled from j to i; this approach relies on link state information and is vulnerable to packet loss and mobility. In Figure 12c, user i transmits a message destined to j with the maximum retransmit distance (MRD) field set to 2; all users with a distance of two or less from j retransmit the message with a MRD set to one less than their distance from j. In Figure 12b, user i transmits the message with MRD set to 3; a wider area is covered and the traffic flows across more paths from i to j, adding additional resiliency. Not shown is an even more resilient configuration of setting MRD to 4; this will cause the top most user to participate in relaying



(a) User j sends a message to the entire group

(b) Only using the reverse path from i to j

(c) Message path when user i sets MRD to 2

(d) Message path when user i sets MRD to 3

Figure 12: Example of a one-to-one traffic flow from user i to j via targeted flooding

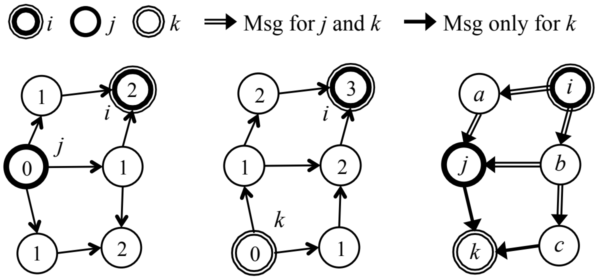
the message, which will allow the packet to travel an even wider path as it moves towards the destination.

3.3.2 *One-to-Many Traffic Pattern*

There are two forms of one-to-many traffic. The first is where one group user desires to send a message to all of the other group members. This traffic pattern is immediately enabled after group discovery is complete, with all relay nodes retransmitting a one-to-many group broadcast message.

The second form of one-to-many traffic is where a message is not intended for the entire group, but still has multiple destinations. This one-to-many traffic pattern is a straightforward extension of the one-to-one presented above. Since there is potentially significant overlap between the paths a message would travel to get to different users, a packet only needs to be retransmitted once instead of sending the same message multiple times over the same set of relays. For increased efficiency, multiple destination/MRD pairs can be specified instead of having a single destination/MRD pair for a message. If a relay hears a message with multiple destination/MRD pairs, it follows the same process as before. If a destination/MRD is no longer valid, then the relay simply drops that destination/MRD pair from the message before retransmitting it.

An example of one-to-some traffic pattern is shown in Figure 13. User i has a packet that it wishes to send to both j and k. In Figure 13a, j sends a one-to-many message, and all users in the group learn their distance to j. Similarly in Figure 13b, all users learn their distance to k. In Figure 13c, user i sends a message destined to both j and k. The packet has two destination/MRD fields: the first is set to $j/1$, and the second is set to $k/2$. Relays a and b retransmit the message and set the MRD field to 0 and 1 for j and k, respectively. Relay c receives the message and sees that the MRD field for destination j is no longer valid; it resends the packet with only k as a destination and an MRD value set to 0. User j will also drop itself as a destination from the message before retransmitting.



(a) Users learn their distance to j

(b) Users learn their distance to k

(c) Message path from i to j and k

Figure 13: A one-to-many traffic flow from user i to both j and k

3.3.3 *Many-to-Many Traffic Pattern*

The many-to-many pattern is implemented as a collection of one-to-many traffic patterns operating jointly. Numerous efficiencies can be gained in the many-to-many traffic pattern by performing coordinated data fusion, source coding, or network coding between the various users and traffic flows [29–32]. Applying these techniques within GCN is a topic of future study.

3.3.4 *Targeted Flooding Performance Evaluation*

We evaluate the performance of targeted flooding by running the following simulation in NS3. 100 users are uniformly distributed in a circular region with a radius of 100 meters. Any user in this region can be a group member with a probability of 25%. Users are either

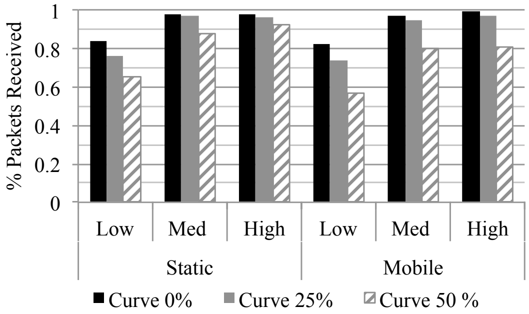


Figure 14: Packet delivery rates using targeted flooding with low, medium, and high resiliency

stationary or move according the the random waypoint model with a constant speed of 2 m/s. The three packet error rate curves that were used in Section 3.2 are used again here. One group member is selected at random to initiate group discovery; this user is labeled the source. Each group member sends a one-to-one message to the source once per second for 100 seconds. The source sends an empty data packet to the group once every two seconds to allow for updated distance information. The number of desired data relays R is set to 3. Three different resiliency values are used for the one-to-one flow: low, medium, and high. When resiliency is low, the MRD is set to one less than that user's distance to its intended destination; i.e., if user i is sending a packet to user j, and user i has distance value $\Delta^i_j = d$, then with low resiliency, the MRD would be set to $d - 1$. With medium, the MRD is set be the same as that user's distance , and with high, the MRD is set to one greater than that user's distance to the destination. The number of packets successfully received at each group member is recorded. Fifty random tests are run, with the results averaged.

Figure 14 shows the results of the simulation. With low resiliency for one-to-one traffic, packet delivery rates range from 84% for `Curve 0%` and no mobility to 57% for `Curve 50%` with mobility. At medium resiliency, packet delivery rates are at 97% or above without mobility for both `Curve 0%` and `Curve 25%`, and at 95% or above with mobility. At high resiliency, packet delivery rates for one-to-one traffic for `Curve 50%` reach 92% without mobility, and are in excess of 80% with mobility. Recall that in Section 2, the link-based routing protocol AODV had one-to-one delivery rates of around 53% for `Curve 0%` and 32% for `Curve 25%`.

4. IMPLEMENTATION AND EVALUATION

In this section, we overview our implementation of Group Centric Networking and perform an evaluation using our implementation of all of the components of GCN working together.

4.1 Implementation

We implemented the full GCN protocol in three environments: simulation, emulation, and hardware. The reason for implementing across all three is to allow us to verify results from one platform against the other, enabling us to have confidence that GCN performs as expected.

To operate in simulation (NS3), we leverage NS3 Direct Code Execution (DCE) [16] which provides a framework to execute existing implementations of userspace and kernelspace network protocols with minimum source code changes. DCE replaces time functions, packet send/receive functions and others with simulation-specific functions to allow NS3 to control input and output to the executing code.

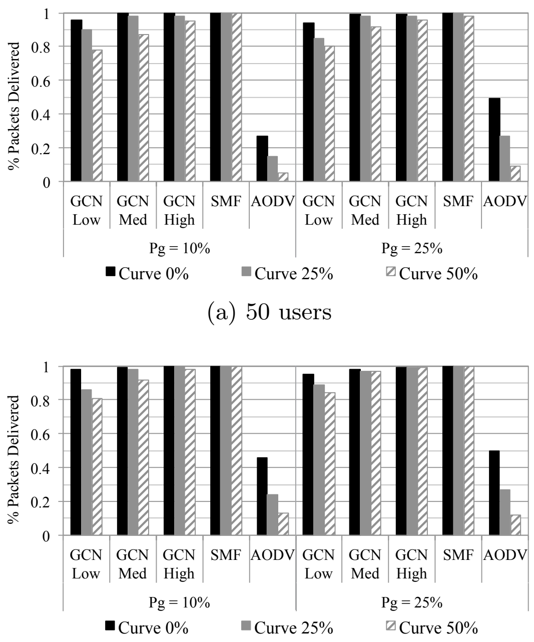
For emulation testing, we leveraged the Extendable Mobile Ad-hoc Network Emulator (EMANE) [33] that emulates layers 1 and 2 (radio and link layers) of the network stack in real-time, and the Common Open Research Emulator (CORE) [18] to help configure, launch, and execute real-time experiments. CORE creates Linux containers that represent network nodes and configures network interfaces, access lists, and processes, which includes the GCN layer. To date, we've successfully validated GCN operation on a 300 node emulation network emulating dozens of hardware platforms.

In addition to emulation, we have successfully tested GCN on actual hardware, with two devices successfully communicating over-the-air using GCN as the network layer. We used a Xilinx Zynq-7000 ZC702 evaluation board to handle the application and network processing, and we used an Analog Devices AD9361 RF transceiver card (that connects directly to the Xilinx evaluation board) for over-the-air transmissions. Implementing GCN on actual hardware provides additional protocol fidelity and verification that GCN can operate on real devices.

4.2 Evaluation of GCN

In this section, we perform an evaluation of a full network configuration using GCN as the network layer. To assess performance, we examine the resiliency and scalability of GCN, and compare GCN against SMF and AODV. In particular, we measure the packet delivery rate and the total amount of bytes sent over-the-air. All tests are performed via simulation using NS3 DCE.

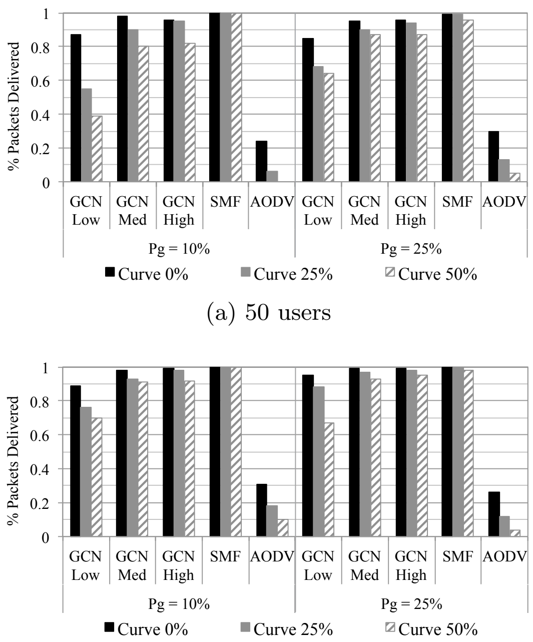
For our tests, we vary the following parameters: number of users, number of group members, packet error rate, and mobility. Users are uniformly distributed in a circular region with a radius of 100 meters. We test both 50 and 100 users in the network, where a user can be a group member with a probability P_g of either 10% or 25%. For packet error rate, the error curves presented in Section 3.2 are used: `Curve 0%`, `Curve 25%`, and `Curve 50%`. For mobility, we test the cases



(a) 50 users

(b) 100 users

Figure 15: Static network: packet delivery rate



(a) 50 users

(b) 100 users

Figure 16: Mobile network: packet delivery rate

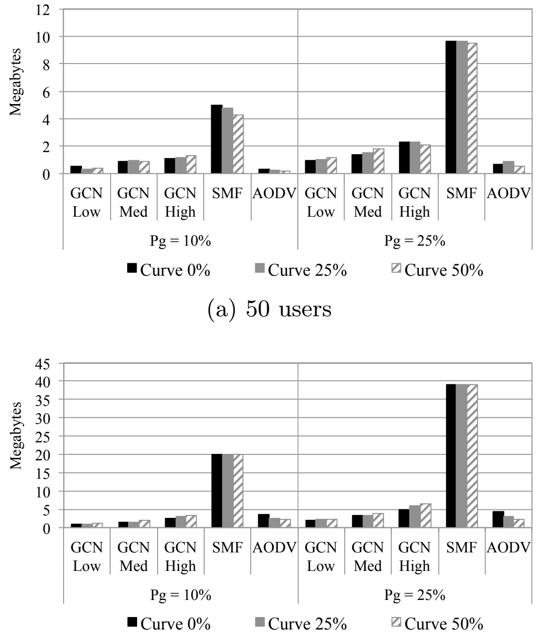
where users are either stationary or move according the the random way point model with zero hold time and a speed that is uniformly selected from 0 to 6 m/s.

The traffic for all scenarios is as follows. A group member is randomly selected as the source, and the source node initiates the group discovery process. The source node transmits one message per second to all other group members via a one-to-many data pattern. All other group members transmit a packet via a many-to-one transmission back to the source node once per second for 100 seconds. The same traffic pattern is run using GCN, SMF, and AODV. Similar to our previous tests, the minimum TTL is selected for SMF such that every group member can reach every other group member, and AODV is run with its default parameters.

We consider three different resiliency levels for GCN: low, medium, and high, where the number of desired data relays R is set to either 3, 6, or 9, respectively. Recall that the parameter R sets the number of additional relays that are selected during the group discovery process. For all three resiliency levels, the maximum retransmit distance (used for the many-to-one traffic pattern) is set to be one greater than a user's distance to its intended destination.

Figure 15 shows the packet delivery rate for a static network. For GCN with low resiliency, approximately 95% of packets are delivered under `Curve 0%` for all combinations of network and group size. This is close to the delivery success rate of SMF, which floods a packet across the network. For `Curve 25%`, GCN is able to deliver 97% of packets using medium resiliency, and under `Curve 50%`, which has a baseline packet error rate of 50%, GCN with high resiliency is able to deliver over 95% of packets for all cases tested. In contrast, AODV is only able to successfully deliver 28% to 50% of packets under `Curve 0%`. The reason for AODV's poor performance under the relatively benign `Curve 0%` is as follows. In `Curve 0%`, short links are error-free and longer links have high error rate. AODV builds a shortest path route to a destination by using the set of exchanged hello messages between users of the network. With sufficiently high frequency, hello messages are successfully exchanged across a high error link, and since this link is of longer distance, it gets used to build a shortest path route. Under `Curve 50%`, where short links are no longer robust due to interference, the delivery rate for AODV ranges from only 6% to 12%.

Figure 16 shows the packet delivery rate for networks with mobile users. As expected, packet delivery rates are lower for all cases tested. Even SMF drops to as low as 95% delivery. AODV now reaches only a maximum of 31% delivery under `Curve 0%` with 100 users, and goes as low as 0% delivery when there are 50 users and $P_g =$ 10% under `Curve 50%`. GCN under low resiliency for `Curve 25%` and `Curve 50%` has has poorer performance for smaller networks and lower group sizes. This is because when there are few users, coverage of the local area under low resiliency is insufficient to adequately provide connectivity for all users in the presence of mobility. Using medium resiliency significantly improves performance for smaller networks and lower group sizes under `Curve 25%` and `Curve 50%`. Using high resiliency in the 50 user network, GCN is able to deliver over 82% of packets under `Curve 50%`, and delivers



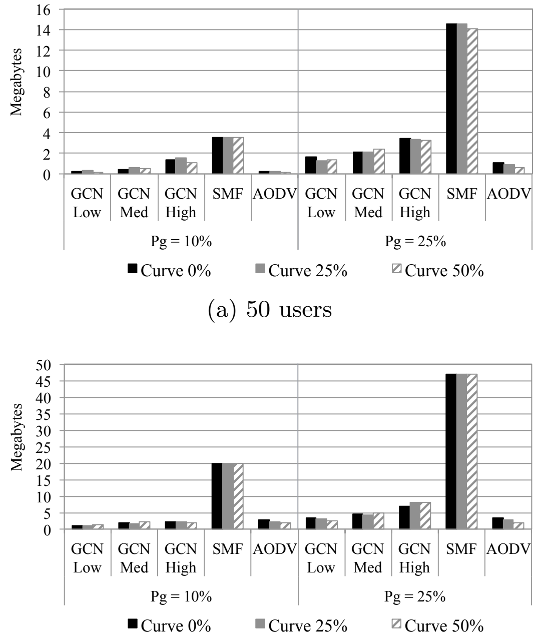
(a) 50 users

(b) 100 users

Figure 17: Static network: over-the-air transmissions

over 95% under `Curve 0%`. In the 100 user network, GCN with high resiliency delivers over 91% of packets under `Curve 50%`, and delivers almost 100% under `Curve 0%`.

While packet delivery is the purpose for any network protocol, future networks of power and bandwidth constrained smart-objects must be able to reliably deliver packets using as few transmissions as possible. Figures 17 and 18 shows the bytes transmitted over-the-air to deliver the traffic sent for the static and mobile scenarios, respectively. While SMF was the most reliable of the the different approaches tested, it came at a very high cost. SMF floods each packet across the network. This is particularly inefficient for the many-to-one traffic pattern, where data will be rebroadcast in areas of the network far from the destination. Additionally, areas with a large number of users will have the same message retransmitted more times than was necessary to all users receive the packet. In contrast, GCN is able to achieve delivery rates comparable to flooding while using an order of magnitude fewer network resources than SMF. GCN is able to selectively choose how many users will relay data in any given area, and is able to keep that number of users relatively constant regardless of the packet error rate being experienced. Furthermore, the many-to-one traffic pattern uses targeted flooding to reliably transmit a packet towards its intended destination, as opposed to SMF that causes each packet to flooded throughout the entire network. GCN allows for highly resilient communication without utilizing significant network resources. AODV and GCN utilize a comparable amount of network resources, but as was shown earlier, AODV is unable to reliably deliver packets in a lossy network. Under `Curve 50%`, AODV had delivery rates ranging from 0% to 12%, while GCN had delivery rates ranging from 82% to 100%.



(a) 50 users

(b) 100 users

Figure 18: Mobile network: over-the-air transmissions

5. CONCLUSION

In this paper, we introduce Group Centric Networking (GCN), which is designed to provide resilient and scalable multi-hop wireless communications for emerging networks of smart-objects. We anticipate that these devices will operate collaboratively as a group in some local region. Hence, the predominant of form of traffic will not be long-distance, but will rather be between members of the group. Additionally, many of these devices will be resource limited and will be operating in a lossy environment. To enable efficient communications under these conditions, GCN is designed to use minimal network resources for data dissemination and to be highly robust to packet loss and mobility.

To verify our approach, and to compare against other wireless networking schemes, we fully implement the GCN protocol in a manner that enables experimentation in NS3, a real-time emulation environment, and on real hardware communicating over-the-air. We find that GCN utilizes up to an order of magnitude fewer network resources than traditional wireless networking schemes, while also achieving superior connectivity and resiliency.

We are currently continuing with research and development for GCN, with areas of study including: using GCN in a multi-channel system, using GCN with systems of directional smart-antennas, additional approaches for resiliency in GCN, and using GCN to support data-centric networking applications.

References

[1] Ericsson, "More than 50 Billion Connected Devices," *White Paper*, February 2011. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf

[2] N. S. Networks, "2020: Beyond 4G Radio Evolution for the Gigabit Experience," *White Paper*, February 2011. [Online]. Available: http://networks.nokia.com/system/files/document/nokia_siemens_networks_beyond_4g_white_paper_online_20082011_0.pdf

[3] C. V. N. Index, "Global mobile data traffic forecast update, 2010-2015," *White Paper, February*, 2011.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with ip: The next internet.* Morgan Kaufmann, 2010.

[6] IETF. (2008) Routing Over Low power and Lossy networks (roll): Charter for Working Group. [Online]. Available: https://datatracker.ietf.org/wg/roll/charter/

[7] M. Dohler, D. Barthel, T. Watteyne, and T. Winter, "Routing requirements for urban low-power and lossy networks," *IETF RFC 5548*, 2009.

[8] K. Pister, P. Thubert, S. Dwars, and T. Phinney, "Industrial routing requirements in low-power and lossy networks," *IETF RFC 5673*, 2009.

[9] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka *et al.*, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *Communications Magazine, IEEE*, vol. 52, no. 5, pp. 26–35, 2014.

[10] T. Watteyne, A. Molinaro, M. G. Richichi, and M. Dohler, "From manet to ietf roll standardization: A paradigm shift in wsn routing protocols," *Communications Surveys & Tutorials, IEEE*, vol. 13, no. 4, pp. 688–707, 2011.

[11] H. Kopetz, "Internet of things," in *Real-Time Systems.* Springer, 2011, pp. 307–323.

[12] A. Brandt and J. Buron, "Home automation routing requirements in low-power and lossy networks," *IETF RFC 5826*, 2010.

[13] J. Martocci, P. Mil, N. Riou, and W. Vermeylen, "Building automation routing requirements in low-power and lossy networks," *IETF RFC 5867*, 2010.

[14] R. Ramanathan, R. Allan, P. Basu, J. Feinberg, G. Jakllari, V. Kawadia, S. Loos, J. Redi, C. Santivanez, and J. Freebersyser, "Scalability of mobile ad hoc networks: Theory vs practice," in *MILCOM 2010.* IEEE, 2010, pp. 493–498.

[15] Network Simulator 3 (NS-3). [Online]. Available: http://www.nsnam.org/

[16] NS3 Direct Code Execution (DCE). [Online]. Available: http://www.nsnam.org/overview/projects/direct-code-execution/

[17] N. Ivanic, B. Rivera, and B. Adamson, "Mobile ad hoc network emulation environment," in *Military Communications Conference, 2009. MILCOM 2009. IEEE.* IEEE, 2009, pp. 1–6.

[18] J. Ahrenholz, T. Goff, and B. Adamson, "Integration of the CORE and EMANE Network Emulators," in *IEEE Military Communications Conference, MILCOM 2011*, 2011.

[19] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[20] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of ieee 802.15.4 using measurements and simulations," in *Wireless communications and networking conference, 2006. WCNC 2006. IEEE*, vol. 1. IEEE, 2006, pp. 487–492.

[21] Bandspeed, "Understanding the effects of radio frequency (rf) interference on wlan performance and security," Tech. Rep., 2010.

[22] A. Hithnawi, H. Shafagh, and S. Duquennoy, "Understanding the impact of cross technology interference on ieee 802.15. 4," in *Proceedings of the 9th ACM international workshop on Wireless network testbeds, experimental evaluation and characterization.* ACM, 2014, pp. 49–56.

[23] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on.* IEEE, 1999, pp. 90–100.

[24] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards," *Computer communications*, vol. 30, no. 7, pp. 1655–1695, 2007.

[25] T. Clausen, J. Yi, and A. C. de Verdiere, "Loadng: Towards aodv version 2," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE.* IEEE, 2012, pp. 1–5.

[26] T. Clausen, A. C. de Verdiere, J. Yi, A. Niktash, Y. Igarashi, H. Satoh, U. Herberg, C. Lavenu, T. Lys, C. Perkins *et al.*, "The lightweight on-demand ad hoc distance-vector routing protocol-next generation (loadng)," *draft-clausen-lln-loadng-12 (work in progress)*, 2014.

[27] O. M. Documentation, "Opnet technologies," *Inc.[Internet] http://www. opnet. com*, 2003.

[28] J. Macker, "Simplified multicast forwarding," *IETF RFC 6621*, 2012.

[29] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," *Information Theory, IEEE Transactions on*, vol. 49, no. 3, pp. 626–643, 2003.

[30] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *Signal Processing Magazine, IEEE*, vol. 21, no. 5, pp. 80–94, 2004.

[31] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 4, p. 24, 2008.

[32] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 243–254.

[33] "Extendable Mobile Ad-Hoc Network Emulator (EMANE)," http://cs.itd.nrl.navy.mil/work/emane/, 2013, [Online, NRL].